

## **Práctica 1 Uso de Git y GitHub**

### **1. Preparación**

Realiza las tareas que se indican a continuación. Incluye las capturas de pantalla que se pidan en un documento tipo LibreOffice o Word.

Cuando se pida realizar un commit recuerda que previamente hay que añadir los archivos al área de preparación si no se ha indicado antes en las instrucciones. En esos casos, un commit significa ejecutar los comandos git add y git commit.

¡IMPORTANTE! No utilices el bloc de notas de Windows para editar los archivos de texto de las tareas. Utiliza en su lugar un editor específico. Algunas sugerencias son:

Visual Studio

Code Atom

Brackets

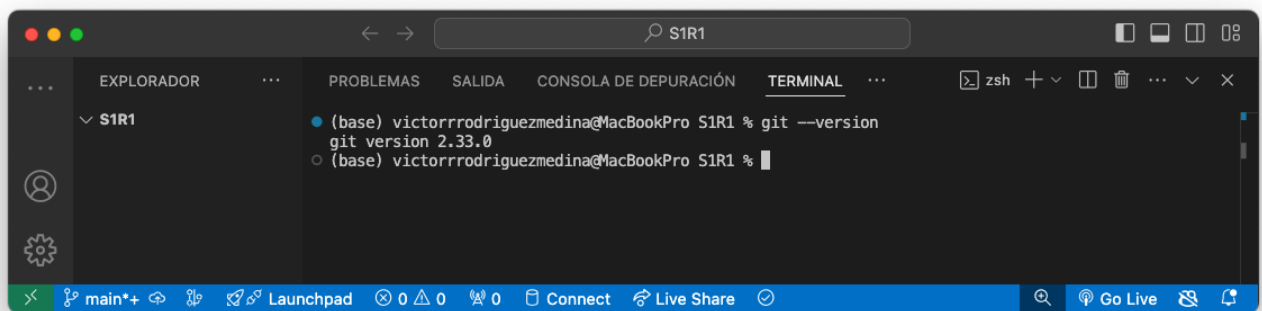
Sublime Text

Notepad ++

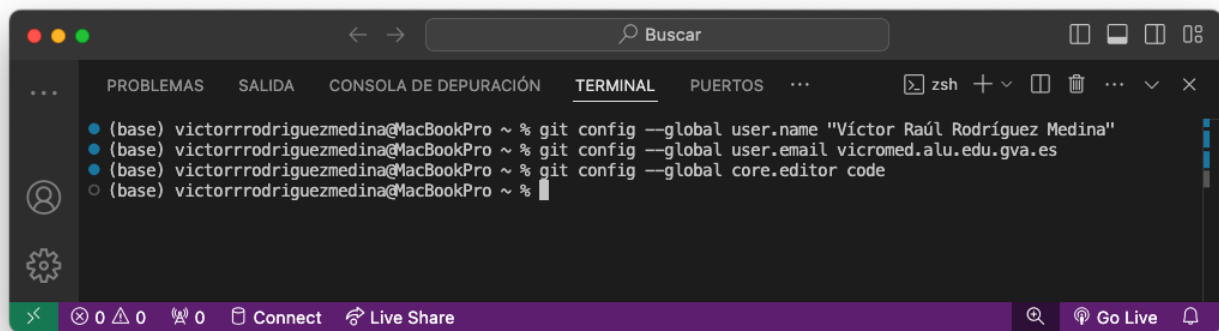
El bloc de notas de Windows utiliza una codificación de caracteres específica de Windows y además incluye la extensión .txt por defecto al final de los nombres de archivo. Por tanto, no es una buena elección para trabajar.

### **2. Fundamentos de Git I**

1. Instala Git en tu sistema operativo. Adjunta una captura de pantalla en la que aparezca el resultado de la ejecución del comando git --version.

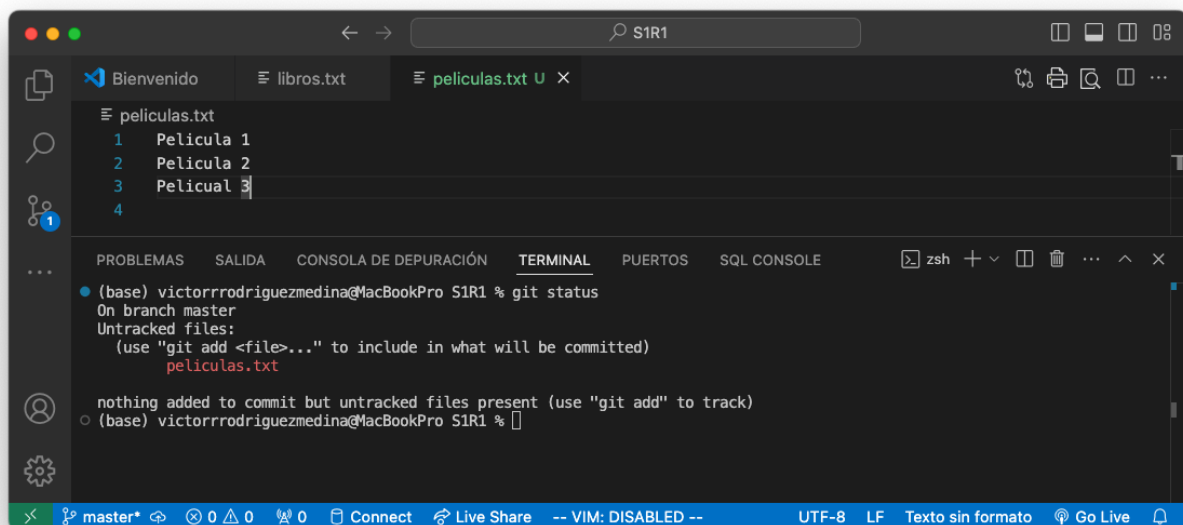


2. Realiza la configuración de Git según lo indicado en el tema (nombre, correo electrónico y editor de preferencia). Adjunta una captura de pantalla con el resultado de la ejecución de los comandos de configuración.



```
(base) victorrrrodriguezmedina@MacBookPro ~ % git config --global user.name "Víctor Raúl Rodríguez Medina"
(base) victorrrrodriguezmedina@MacBookPro ~ % git config --global user.email vicromed.alu.edu.gva.es
(base) victorrrrodriguezmedina@MacBookPro ~ % git config --global core.editor code
(base) victorrrrodriguezmedina@MacBookPro ~ %
```

3. Crea una carpeta denominada S1R1. Realiza las siguientes acciones en ella:
- Crea un repositorio Git.
  - Crea un fichero denominado libros.txt. Añade tres títulos de libros cada uno en una línea distinta.
  - Haz un primer commit
  - Añade dos libros al archivo libros.txt.
  - Haz un segundo commit.
  - Crea un fichero denominado peliculas.txt. Añade tres títulos de películas a dicho archivo.
  - Haz una captura de pantalla del comando git status.

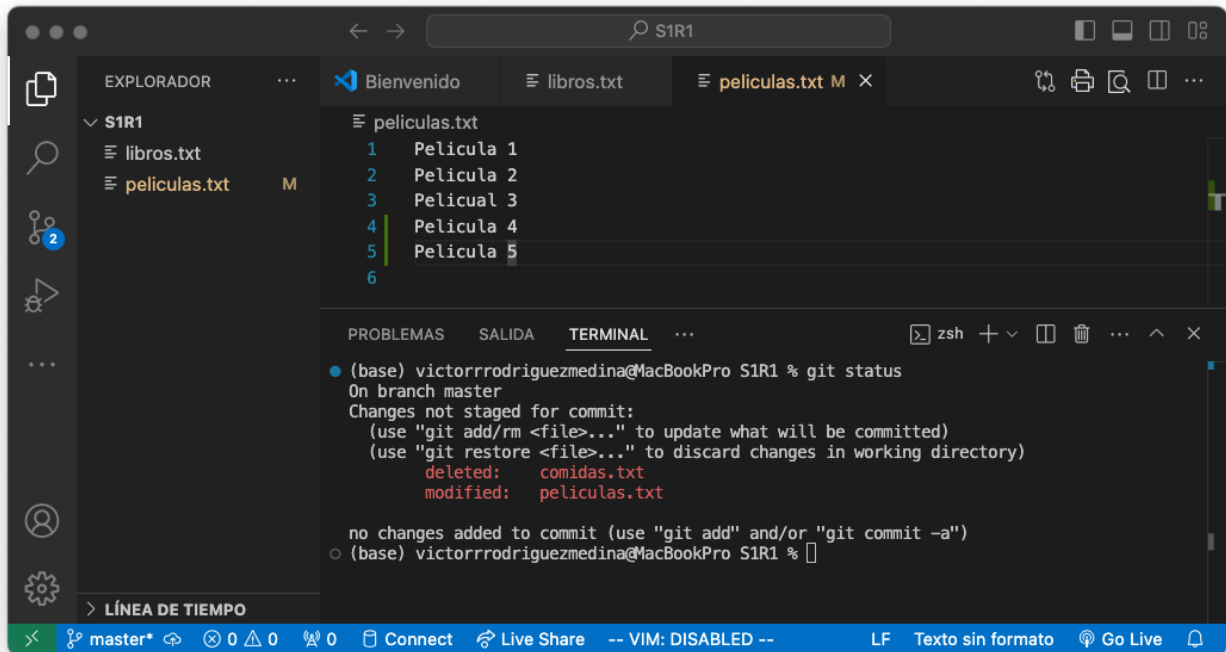


```
películas.txt
1 Película 1
2 Película 2
3 Película 3
4

(base) victorrrrodriguezmedina@MacBookPro S1R1 % git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    películas.txt

nothing added to commit but untracked files present (use "git add" to track)
(base) victorrrrodriguezmedina@MacBookPro S1R1 %
```

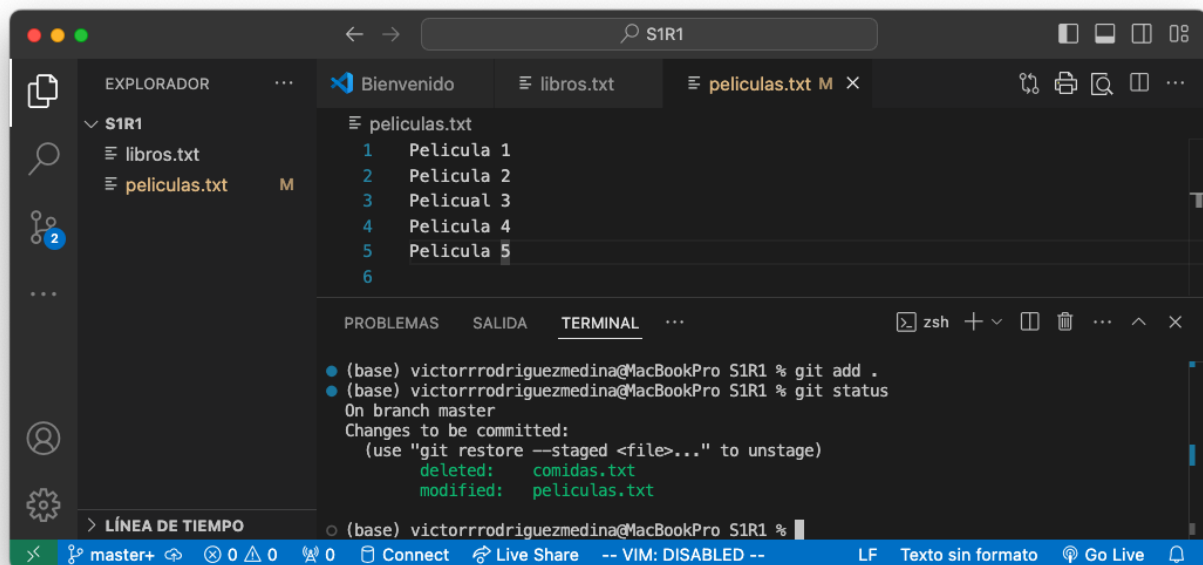
- Crea un fichero denominado comidas.txt. Añade tres nombres de comidas a dicho archivo.
- Haz un tercer commit que incluya los archivos peliculas.txt y comidas.txt.
- Elimina el archivo comidas.txt desde el navegador de archivos.
- Añade dos películas más al archivo peliculas.txt.
- Haz una captura de pantalla que muestre los cambios en el directorio de trabajo.



```
(base) victorrrrodriguezmedina@MacBookPro S1R1 % git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    comidas.txt
    modified:   películas.txt

no changes added to commit (use "git add" and/or "git commit -a")
(base) victorrrrodriguezmedina@MacBookPro S1R1 %
```

- m) Añade los cambios al área de preparación.
- n) Haz una captura de pantalla del comando git status. Debe indicar que se ha borrado el archivo comidas.txt y que se ha modificado el archivo películas.txt.

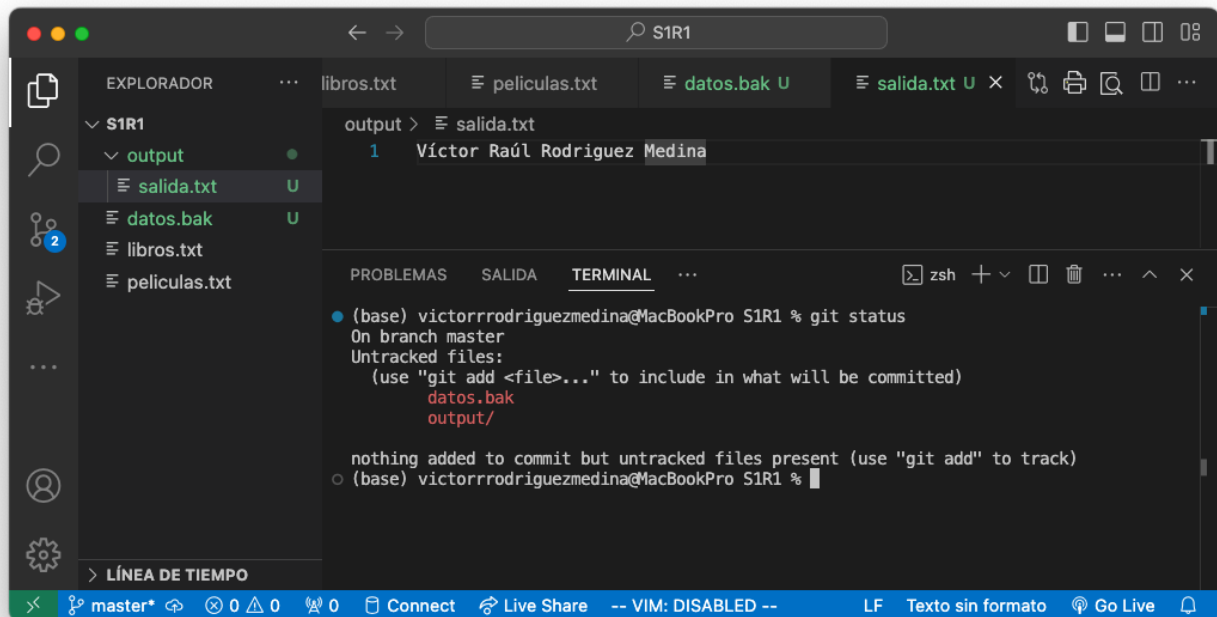


```
(base) victorrrrodriguezmedina@MacBookPro S1R1 % git add .
(base) victorrrrodriguezmedina@MacBookPro S1R1 % git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    comidas.txt
    modified:   películas.txt

(base) victorrrrodriguezmedina@MacBookPro S1R1 %
```

- ñ) Haz un cuarto commit.
- o) Crea un archivo denominado datos.bak. Añade tres títulos de libros a dicho archivo. ¡IMPORTANTE! No añadas el archivo al área de preparación ni hagas ningún commit.
- p) Crea una subcarpeta denominada output. Crea un archivo denominado salida.txt en su interior. Escribe tu nombre y apellidos en dicho archivo. ¡IMPORTANTE! No añadas los archivos al área de preparación ni hagas ningún commit.

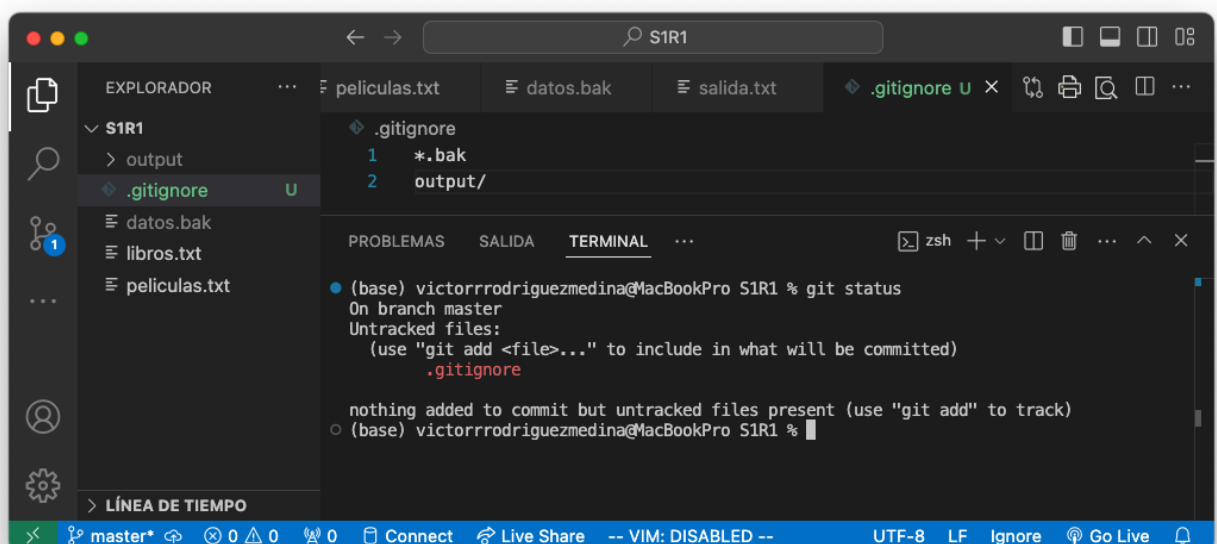
q) Haz una captura de pantalla del comando `git status`. Deben aparecer los archivos `datos.bak` y `output/salida.txt` como archivos nuevos (color rojo).



```
(base) victorrrrodriguezmedina@MacBookPro S1R1 % git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    datos.bak
    output/

nothing added to commit but untracked files present (use "git add" to track)
(base) victorrrrodriguezmedina@MacBookPro S1R1 %
```

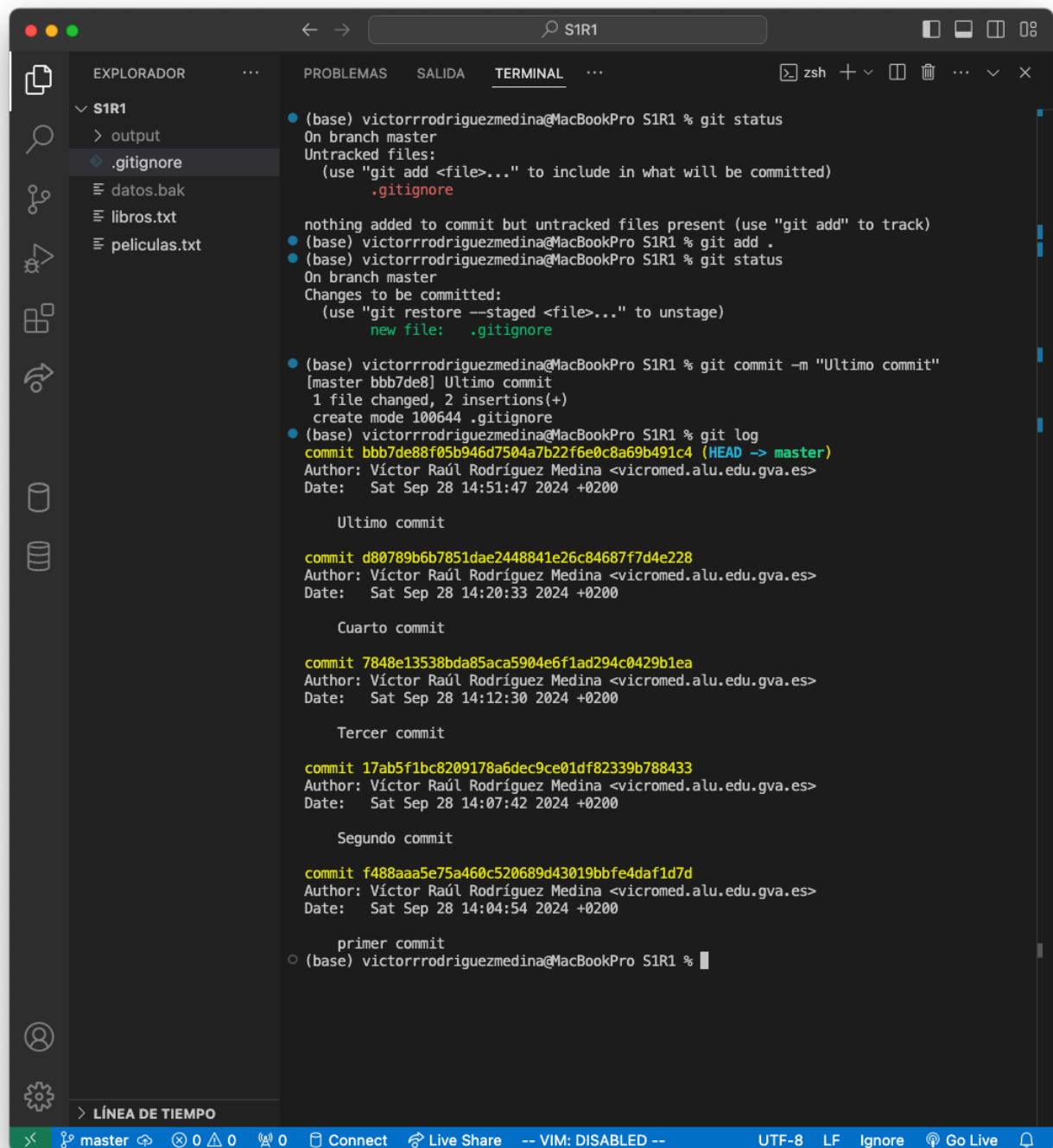
- r) Crea un archivo `.gitignore` para que los ficheros con extensión `.bak` y el contenido de la carpeta `output/` no se incluyan en el repositorio.
- s) Haz una nueva captura de pantalla del comando `git status`. Ahora no deben aparecer los archivos `datos.bak` y `output/salida.txt` como archivos nuevos, sino que en su lugar debe aparecer únicamente el archivo `.gitignore`.



```
(base) victorrrrodriguezmedina@MacBookPro S1R1 % git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

nothing added to commit but untracked files present (use "git add" to track)
(base) victorrrrodriguezmedina@MacBookPro S1R1 %
```

- t) Haz un último commit para incluir el archivo `.gitignore` en el repositorio.
- u) Haz una captura de pantalla que muestre el histórico de cambios del repositorio.



```
(base) victorrrrodriguezmedina@MacBookPro S1R1 % git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

nothing added to commit but untracked files present (use "git add" to track)
(base) victorrrrodriguezmedina@MacBookPro S1R1 % git add .
(base) victorrrrodriguezmedina@MacBookPro S1R1 % git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   .gitignore

(base) victorrrrodriguezmedina@MacBookPro S1R1 % git commit -m "Ultimo commit"
[master bbb7de8] Ultimo commit
1 file changed, 2 insertions(+)
create mode 100644 .gitignore
(base) victorrrrodriguezmedina@MacBookPro S1R1 % git log
commit bbb7de8f05b946d7504a7b22f6e0c8a69b491c4 (HEAD -> master)
Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
Date: Sat Sep 28 14:51:47 2024 +0200

    Ultimo commit

commit d80789b6b7851dae2448841e26c84687f7d4e228
Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
Date: Sat Sep 28 14:20:33 2024 +0200

    Cuarto commit

commit 7848e13538bda85aca5904e6f1ad294c0429b1ea
Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
Date: Sat Sep 28 14:12:30 2024 +0200

    Tercer commit

commit 17ab5f1bc8209178a6dec9ce01df82339b788433
Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
Date: Sat Sep 28 14:07:42 2024 +0200

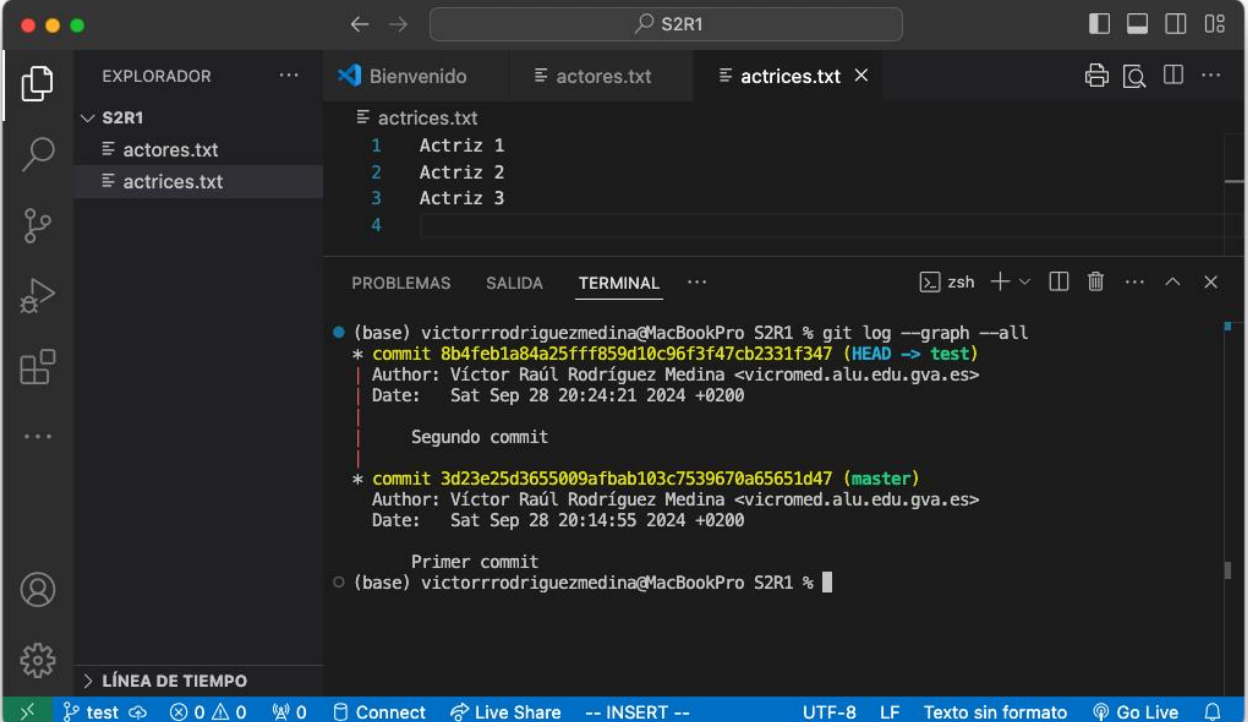
    Segundo commit

commit f488aaa5e75a460c520689d43019bbfe4daf1d7d
Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
Date: Sat Sep 28 14:04:54 2024 +0200

    primer commit
(base) victorrrrodriguezmedina@MacBookPro S1R1 %
```

### 3. Fundamentos de Git II

1. Crea una carpeta denominada S2R1. Realiza las siguientes acciones en ella:
  - a) Crea un repositorio Git.
  - b) Crea un fichero denominado actores.txt. Añade tres nombres de actores cada uno en una línea distinta.
  - c) Haz un primer commit.
  - d) Crea una rama denominada test.
  - e) Cambia a la rama test
  - f) En la rama test crea un fichero denominado actrices.txt. Añade tres nombres de actrices y realiza un commit en dicha rama.
  - g) Haz una captura de pantalla del resultado del comando `git log --graph --all`.



```
(base) victorrrrodriguezmedina@MacBookPro S2R1 % git log --graph --all
* commit 8b4feb1a84a25fff859d10c96f3f47cb2331f347 (HEAD -> test)
  Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
  Date: Sat Sep 28 20:24:21 2024 +0200

    Segundo commit

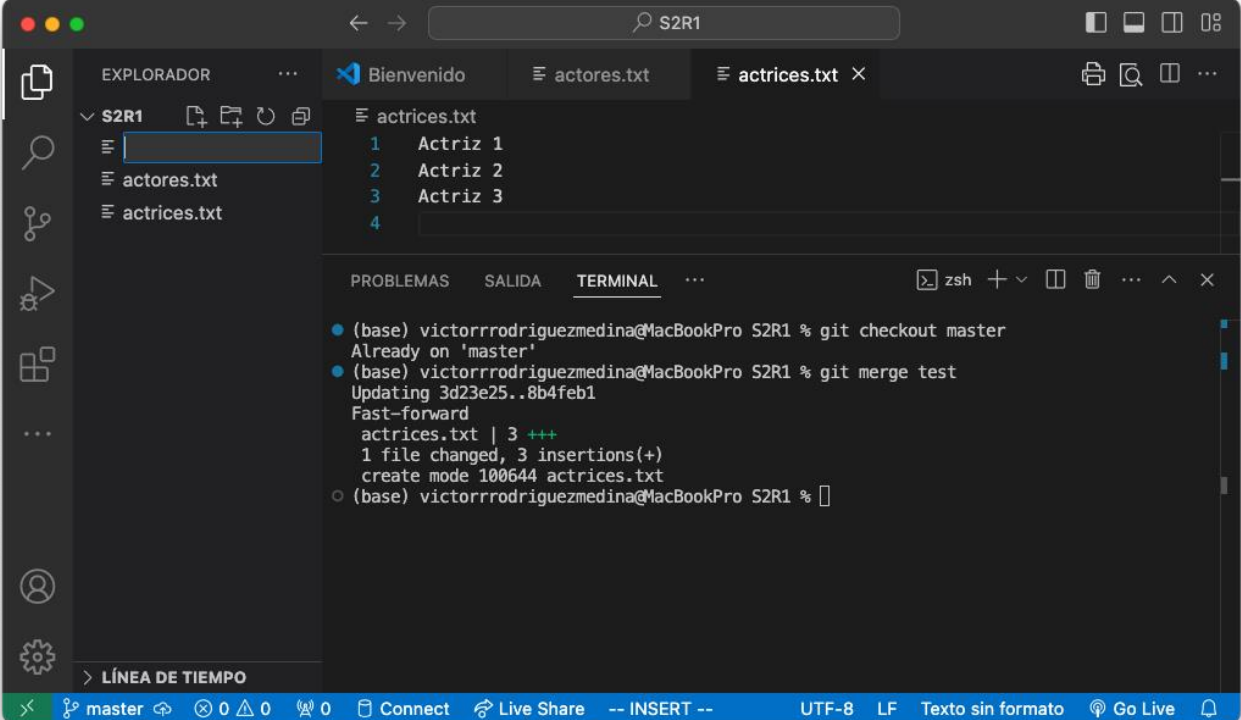
* commit 3d23e25d3655009afbab103c7539670a65651d47 (master)
  Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
  Date: Sat Sep 28 20:14:55 2024 +0200

    Primer commit

○ (base) victorrrrodriguezmedina@MacBookPro S2R1 %
```

- h) Cambia a la rama master.
- i) Incorpora los cambios de la rama test a la rama master. Haz una captura de pantalla de los comandos que has utilizado y de su resultado.





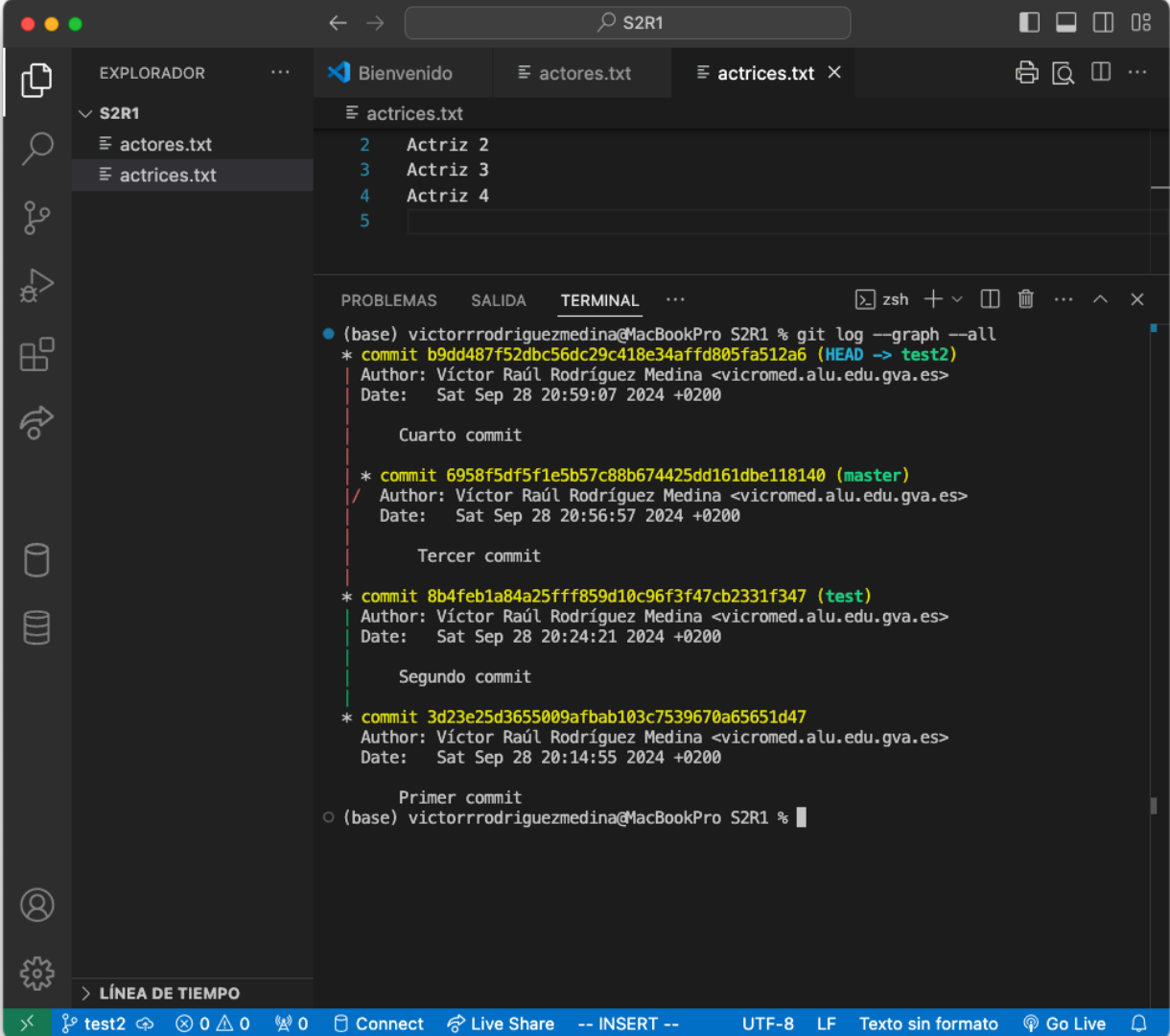
The screenshot shows a VS Code editor window with a terminal at the bottom. The terminal displays the following output:

```
(base) victorrrrodriguezmedina@MacBookPro S2R1 % git checkout master
Already on 'master'
(base) victorrrrodriguezmedina@MacBookPro S2R1 % git merge test
Updating 3d23e25..8b4feb1
Fast-forward
 actrices.txt | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 actrices.txt
(base) victorrrrodriguezmedina@MacBookPro S2R1 %
```

The editor shows a file named `actrices.txt` with the following content:

```
1 Actriz 1
2 Actriz 2
3 Actriz 3
4
```

- j) Crea una segunda rama denominada test2. La rama test2 apunta al mismo commit que la rama master en este momento.
- k) En la rama master, añade una actriz al fichero `actrices.txt` y haz un commit.
- l) Cambia a la rama test2
- m) En la rama test2, añade una actriz al fichero `actrices.txt` y haz otro commit.
- n) Haz una captura de pantalla del resultado del comando `git log --graph --all`. Debe haber dos caminos distintos: uno para la rama master y otro para la rama test2.



The screenshot shows a Visual Studio Code editor window. On the left is the Explorer sidebar showing a project named 'S2R1' with two files: 'actores.txt' and 'actrices.txt'. The 'actrices.txt' file is open in the editor, showing three lines of text: 'Actriz 2', 'Actriz 3', and 'Actriz 4'. Below the editor is a terminal window with the title 'TERMINAL'. It shows the output of the command 'git log --graph --all'. The output lists five commits, with the most recent being the HEAD (test2). The commits are labeled as 'Primer commit', 'Segundo commit', 'Tercer commit', 'Cuarto commit', and 'Quinto commit'. The terminal output is as follows:

```
(base) victorrrrodriguezmedina@MacBookPro S2R1 % git log --graph --all
* commit b9dd487f52dbc56dc29c418e34affd805fa512a6 (HEAD -> test2)
  Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
  Date: Sat Sep 28 20:59:07 2024 +0200

    Cuarto commit

* commit 6958f5df5f1e5b57c88b674425dd161dbe118140 (master)
  Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
  Date: Sat Sep 28 20:56:57 2024 +0200

    Tercer commit

* commit 8b4feb1a84a25fff859d10c96f3f47cb2331f347 (test)
  Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
  Date: Sat Sep 28 20:24:21 2024 +0200

    Segundo commit

* commit 3d23e25d3655009afbab103c7539670a65651d47
  Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
  Date: Sat Sep 28 20:14:55 2024 +0200

    Primer commit
○ (base) victorrrrodriguezmedina@MacBookPro S2R1 %
```

ñ) Cambia a la rama master

o) Incorpora los cambios de la rama test2 a la rama master. ¿Se produce un conflicto? De ser así realiza una captura del comando git status.



```
EXPLORADOR  ...  Bienvenido  actores.txt  actrices.txt x
v S2R1
  actores.txt
  actrices.txt

2 Actriz 2
3 Actriz 3
4 Actriz 4
5

PROBLEMAS  SALIDA  TERMINAL  ...
zsh + v  [icon]  [icon]  [icon]  [icon]

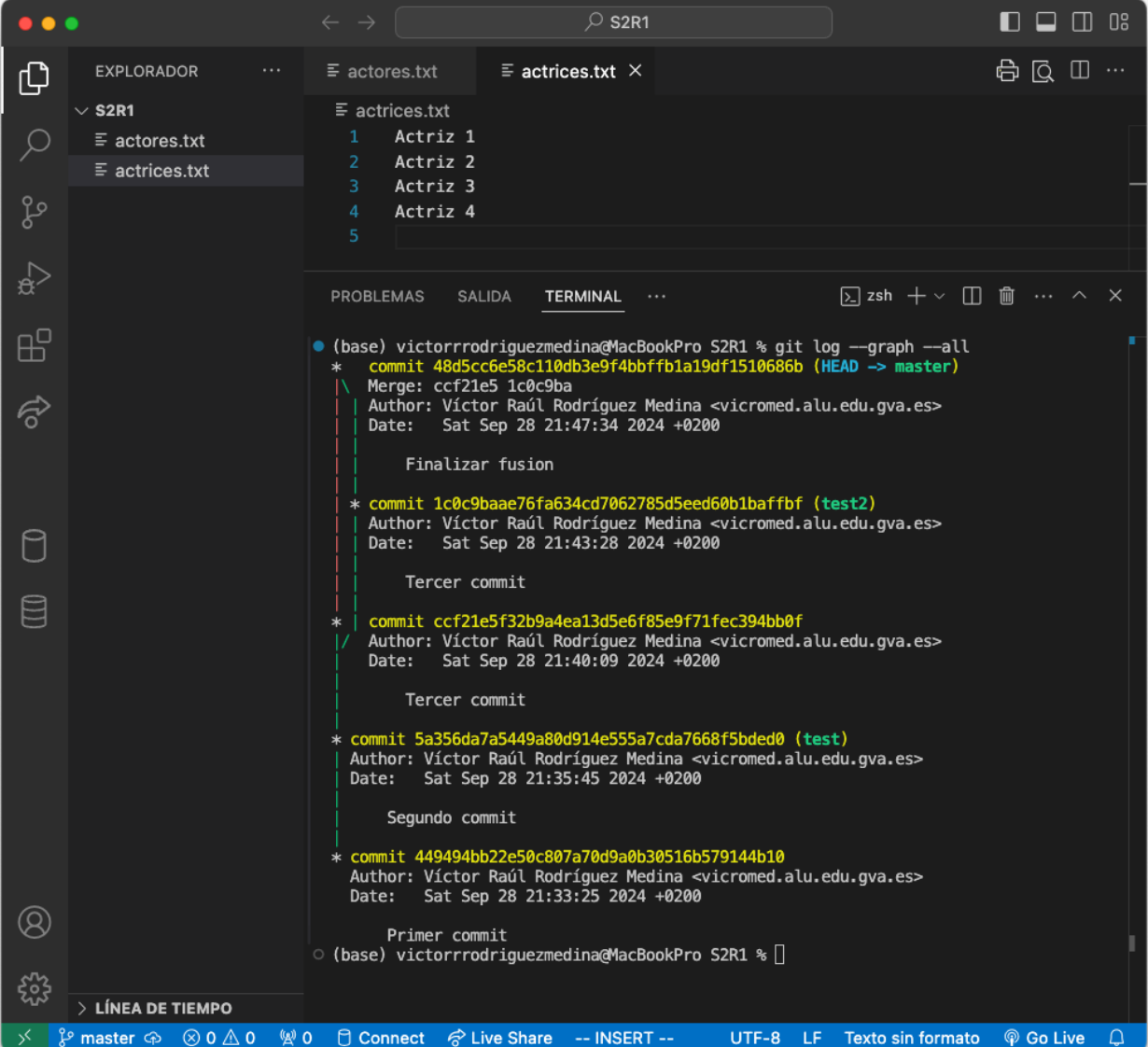
• (base) victorrrrodriguezmedina@MacBookPro S2R1 % git checkout master
Switched to branch 'master'
• (base) victorrrrodriguezmedina@MacBookPro S2R1 % git merge test2
hint: Waiting for your editor to close the file... code --wait: code: command not fou
nd
error: There was a problem with the editor 'code --wait'.
Not committing merge; use 'git commit' to complete the merge.
• (base) victorrrrodriguezmedina@MacBookPro S2R1 % git status
On branch master
All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

○ (base) victorrrrodriguezmedina@MacBookPro S2R1 %

> LÍNEA DE TIEMPO
x  master  0 0 0  Connect  Live Share  -- INSERT --  UTF-8  LF  Texto sin formato  Go Live  [icon]
```

p) Resuelve el conflicto incorporando los dos nombres de actrices.

q) Haz una captura de pantalla del resultado del comando `git log --graph --all`. Observa que se ha creado un nuevo commit que integra los dos caminos anteriores.



The screenshot shows a code editor interface with a sidebar on the left labeled 'EXPLORADOR' containing a file tree for 'S2R1' with files 'actores.txt' and 'actrices.txt'. The main editor area has two tabs: 'actores.txt' and 'actrices.txt'. The 'actrices.txt' tab is active, showing a list of names: 'Actriz 1', 'Actriz 2', 'Actriz 3', 'Actriz 4', and an empty line. Below the editor is a terminal window titled 'TERMINAL' with a 'zsh' shell. The terminal output shows the result of the command 'git log --graph --all', displaying a commit graph with five commits. The commits are: a merge commit (48d5cc6e58c110db3e9f4bbffb1a19df1510686b) merging 'HEAD' into 'master'; a commit (1c0c9baae76fa634cd7062785d5eed60b1baffbf) labeled '(test2)'; a commit (ccf21e5f32b9a4ea13d5e6f85e9f71fec394bb0f) labeled '(test)'; a commit (5a356da7a5449a80d914e555a7cda7668f5bde0) labeled '(test)'; and a commit (449494bb22e50c807a70d9a0b30516b579144b10) labeled '(test)'. The terminal also shows the command 'git log --graph --all' and the output of 'git log --graph --all'.

```
(base) victorrrrodriguezmedina@MacBookPro S2R1 % git log --graph --all
* commit 48d5cc6e58c110db3e9f4bbffb1a19df1510686b (HEAD -> master)
  Merge: ccf21e5 1c0c9ba
  Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
  Date: Sat Sep 28 21:47:34 2024 +0200

    Finalizar fusion

* commit 1c0c9baae76fa634cd7062785d5eed60b1baffbf (test2)
  Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
  Date: Sat Sep 28 21:43:28 2024 +0200

    Tercer commit

* commit ccf21e5f32b9a4ea13d5e6f85e9f71fec394bb0f
  Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
  Date: Sat Sep 28 21:40:09 2024 +0200

    Tercer commit

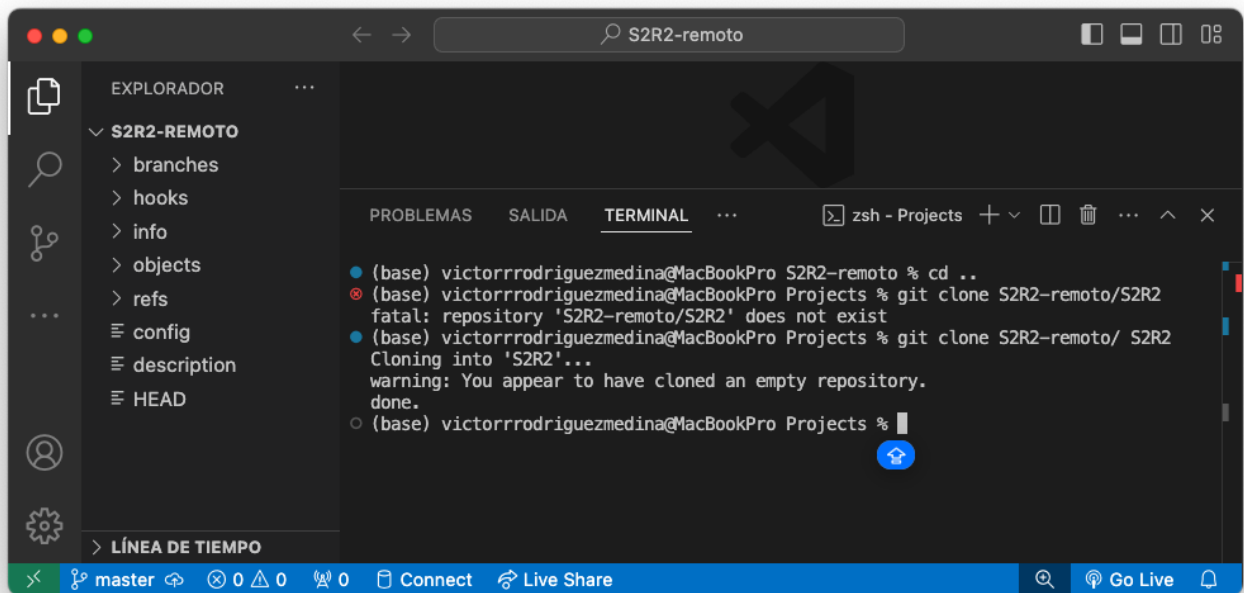
* commit 5a356da7a5449a80d914e555a7cda7668f5bde0 (test)
  Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
  Date: Sat Sep 28 21:35:45 2024 +0200

    Segundo commit

* commit 449494bb22e50c807a70d9a0b30516b579144b10
  Author: Víctor Raúl Rodríguez Medina <vicromed.alu.edu.gva.es>
  Date: Sat Sep 28 21:33:25 2024 +0200

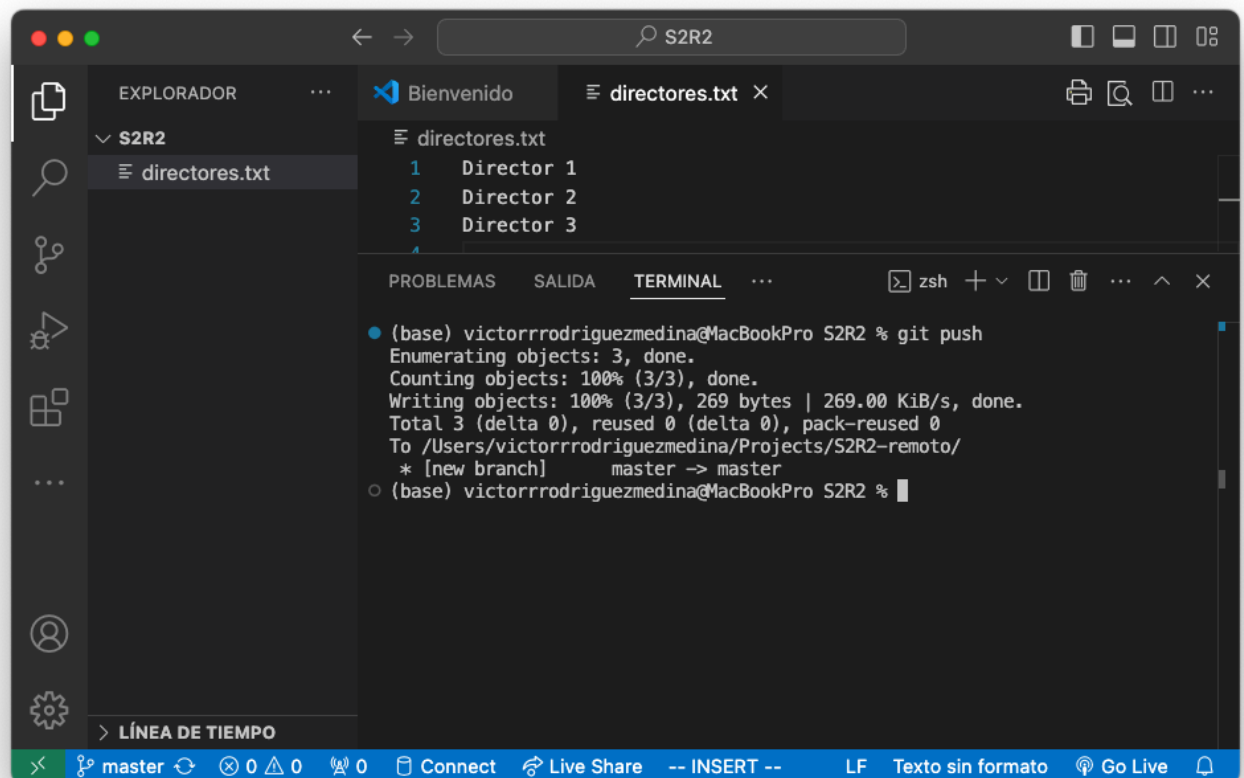
    Primer commit
o (base) victorrrrodriguezmedina@MacBookPro S2R1 %
```

2. Crea una carpeta denominada S2R2-remoto. Inicializa un repositorio Git en su interior mediante el comando `git init --bare`. Esta carpeta se utilizará como repositorio remoto.
3. Clona el repositorio S2R2-remoto en una carpeta denominada S2R2. Adjunta captura de pantalla del resultado del comando de clonado. A continuación, realiza las siguientes acciones en el repositorio S2R2:



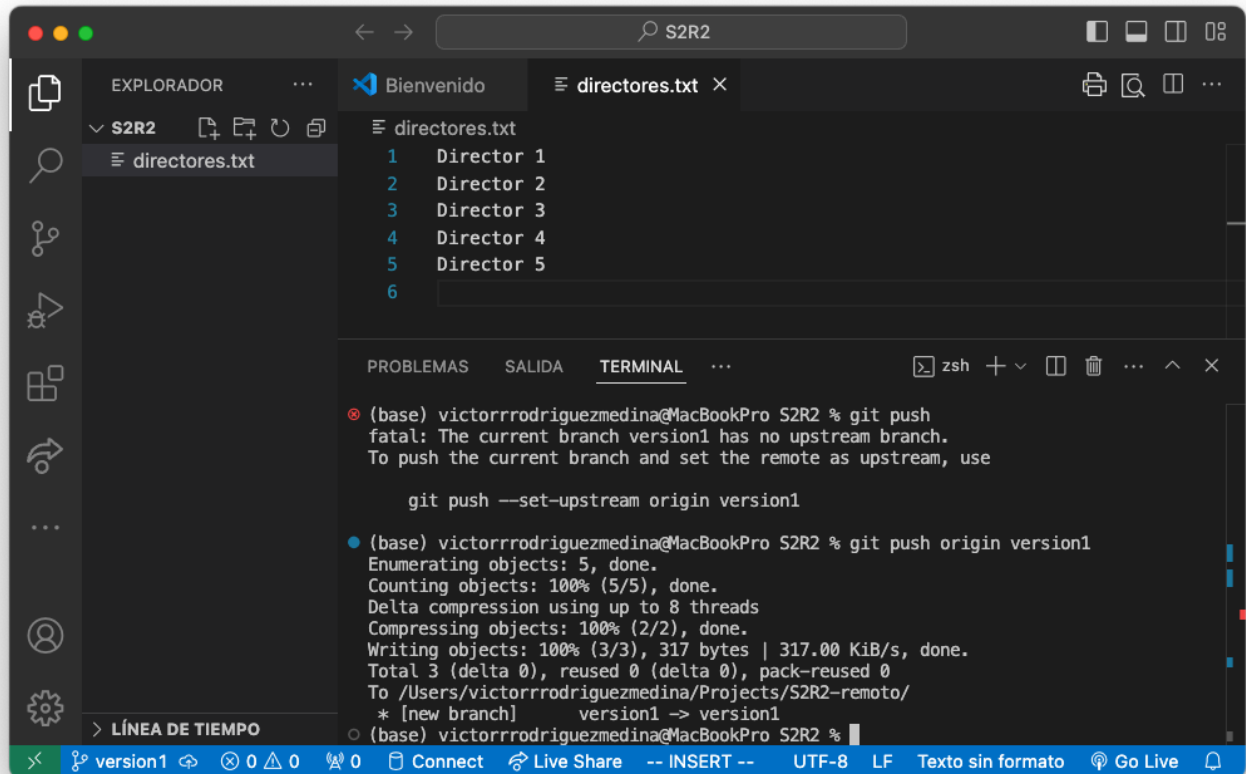
```
(base) victorrrrodriguezmedina@MacBookPro S2R2-remoto % cd ..
(base) victorrrrodriguezmedina@MacBookPro Projects % git clone S2R2-remoto/S2R2
fatal: repository 'S2R2-remoto/S2R2' does not exist
(base) victorrrrodriguezmedina@MacBookPro Projects % git clone S2R2-remoto/ S2R2
Cloning into 'S2R2'...
warning: You appear to have cloned an empty repository.
done.
(base) victorrrrodriguezmedina@MacBookPro Projects %
```

- Crea un archivo denominado `directores.txt`. Añade el nombre de tres directores de cine.
- Haz un commit.
- Realiza un push al repositorio remoto. Adjunta captura de pantalla del resultado.



```
(base) victorrrrodriguezmedina@MacBookPro S2R2 % git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 269 bytes | 269.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To /Users/victorrrrodriguezmedina/Projects/S2R2-remoto/
 * [new branch]      master -> master
(base) victorrrrodriguezmedina@MacBookPro S2R2 %
```

- d) Crea una rama denominada version1.
- e) Cambia a la rama version1.
- f) En la rama version1 añade el nombre de dos directores de cine más al archivo directores.txt y haz un commit de los cambios.
- g) Realiza un push de la rama al repositorio remoto de manera que quede asociada a la rama remota del mismo nombre. Adjunta captura de pantalla del resultado.



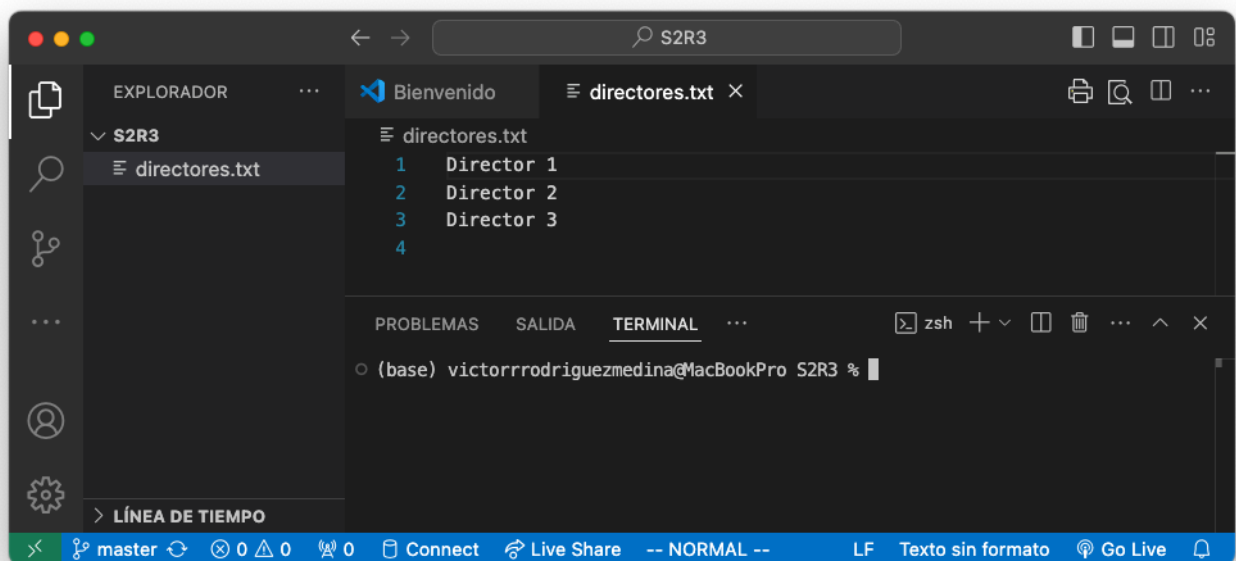
```
(base) victorrrrodriguezmedina@MacBookPro S2R2 % git push
fatal: The current branch version1 has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin version1

(base) victorrrrodriguezmedina@MacBookPro S2R2 % git push origin version1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 317 bytes | 317.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To /Users/victorrrrodriguezmedina/Projects/S2R2-remoto/
 * [new branch]      version1 -> version1
(base) victorrrrodriguezmedina@MacBookPro S2R2 %
```

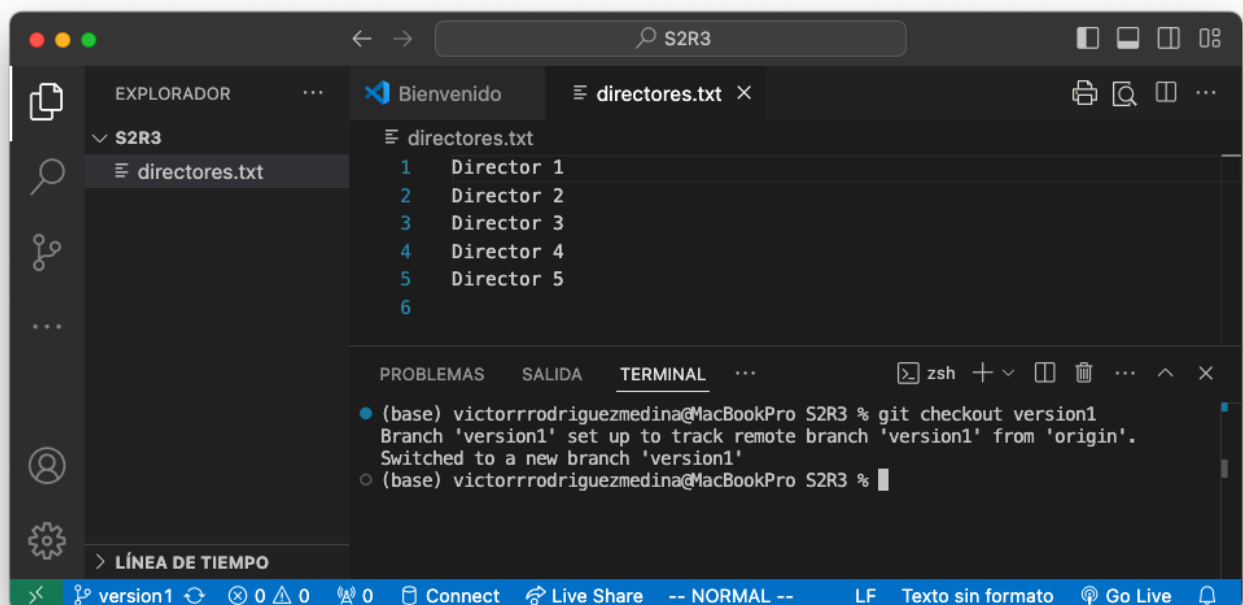
4. Clona el repositorio S2R2-remoto en una segunda carpeta denominada S2R3. Realiza las siguientes acciones sobre ella:

- a) Muestra en la consola el contenido del fichero directores.txt y el resultado del comando git status. Debe mostrar tres directores.



b) Cambia a la rama `version1`. Muestra el resultado del comando. Comprueba que se crea una rama local `version1` con el contenido de la rama remota `origin/version1` y enlazada con ella. Al clonar el repositorio la rama no existía (solo se clona la rama principal, `master`), pero al cambiar a una rama que existe en el remoto se produce su creación local y enlazado con su correspondiente remota.

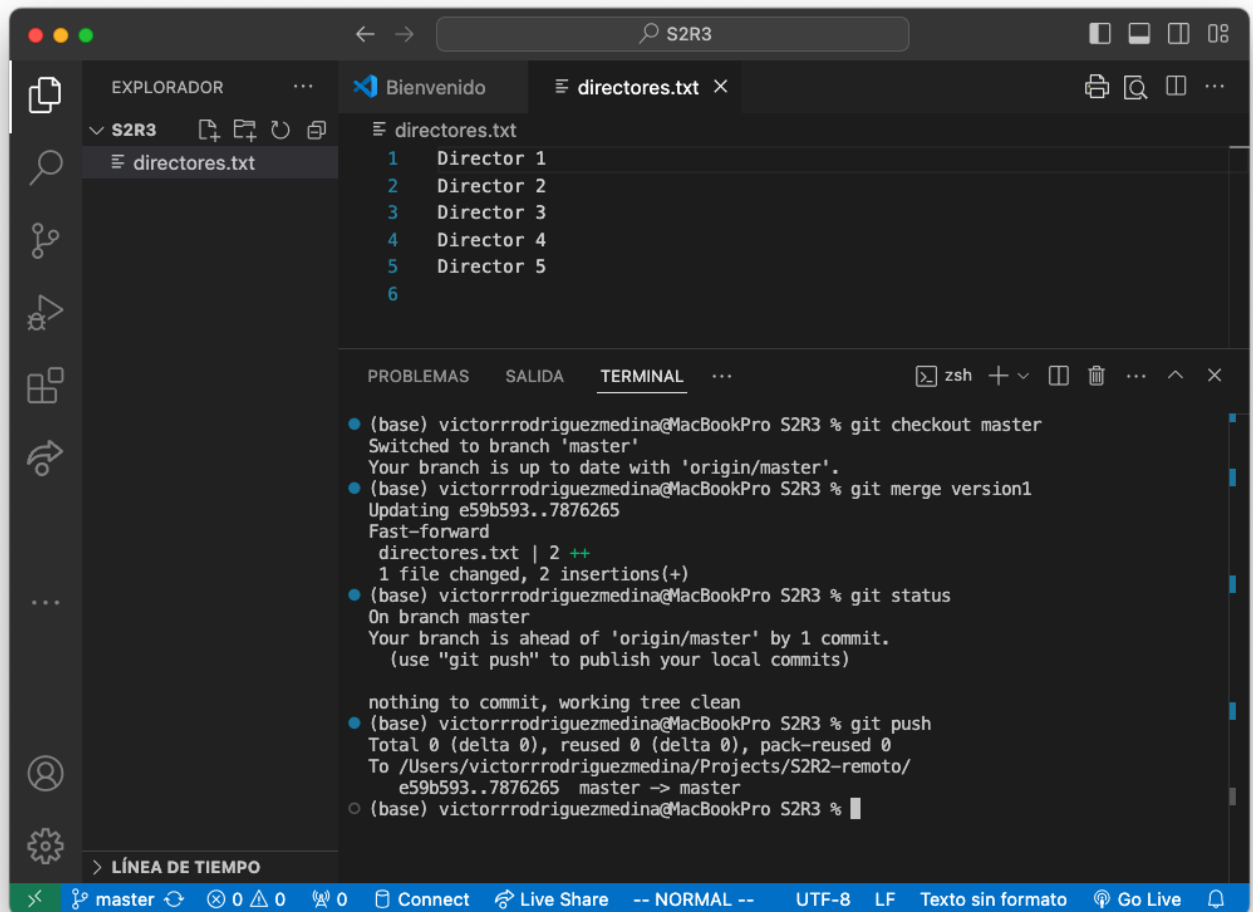
c) Muestra el contenido del fichero `directores.txt` por la pantalla. Comprueba que se muestran los 5 nombres de directores esperados. Adjunta captura de pantalla.



d) Cambia a la rama `master`.

e) Incorpora los cambios de la rama `version1` a la rama `master`.

f) Sube la rama master actualizada al servidor. Adjunta captura de pantalla del resultado del comando.



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

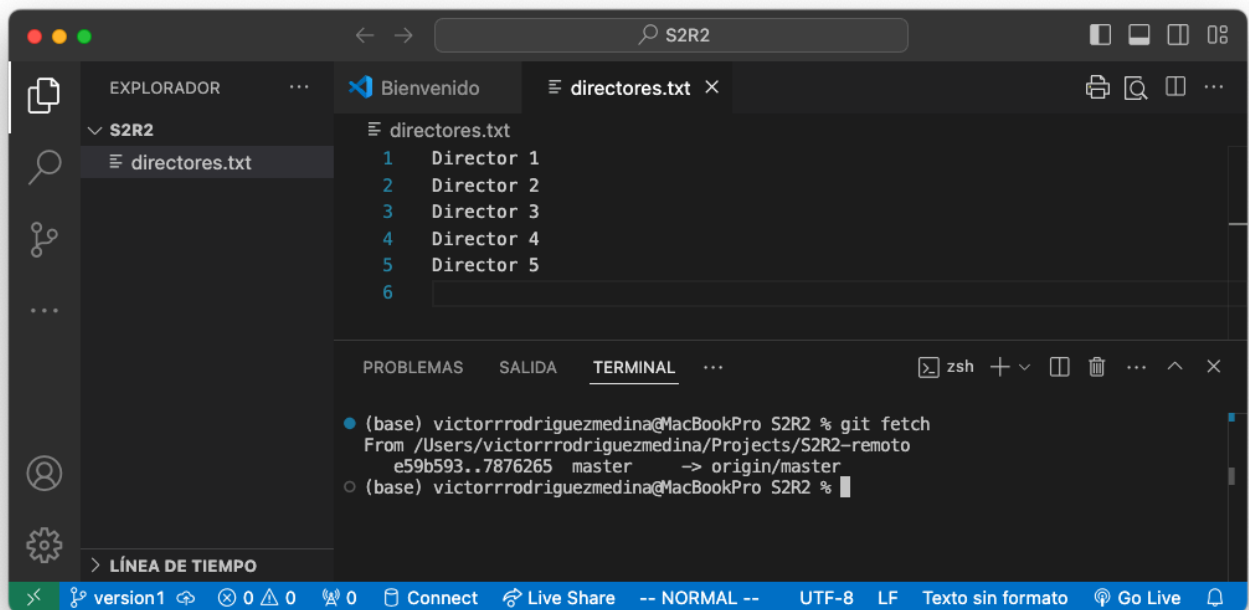
```
(base) victorrrrodriguezmedina@MacBookPro S2R3 % git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
(base) victorrrrodriguezmedina@MacBookPro S2R3 % git merge version1
Updating e59b593..7876265
Fast-forward
 directores.txt | 2 ++
 1 file changed, 2 insertions(+)
(base) victorrrrodriguezmedina@MacBookPro S2R3 % git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
(base) victorrrrodriguezmedina@MacBookPro S2R3 % git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To /Users/victorrrrodriguezmedina/Projects/S2R2-remoto/
 e59b593..7876265 master -> master
(base) victorrrrodriguezmedina@MacBookPro S2R3 %
```

The Explorer view on the left shows the file structure of the S2R3 project, with the file `directores.txt` selected. The file content is visible in the editor, showing a list of directors.

5. Vuelve de nuevo a la carpeta S2R2 y realiza las siguientes acciones:

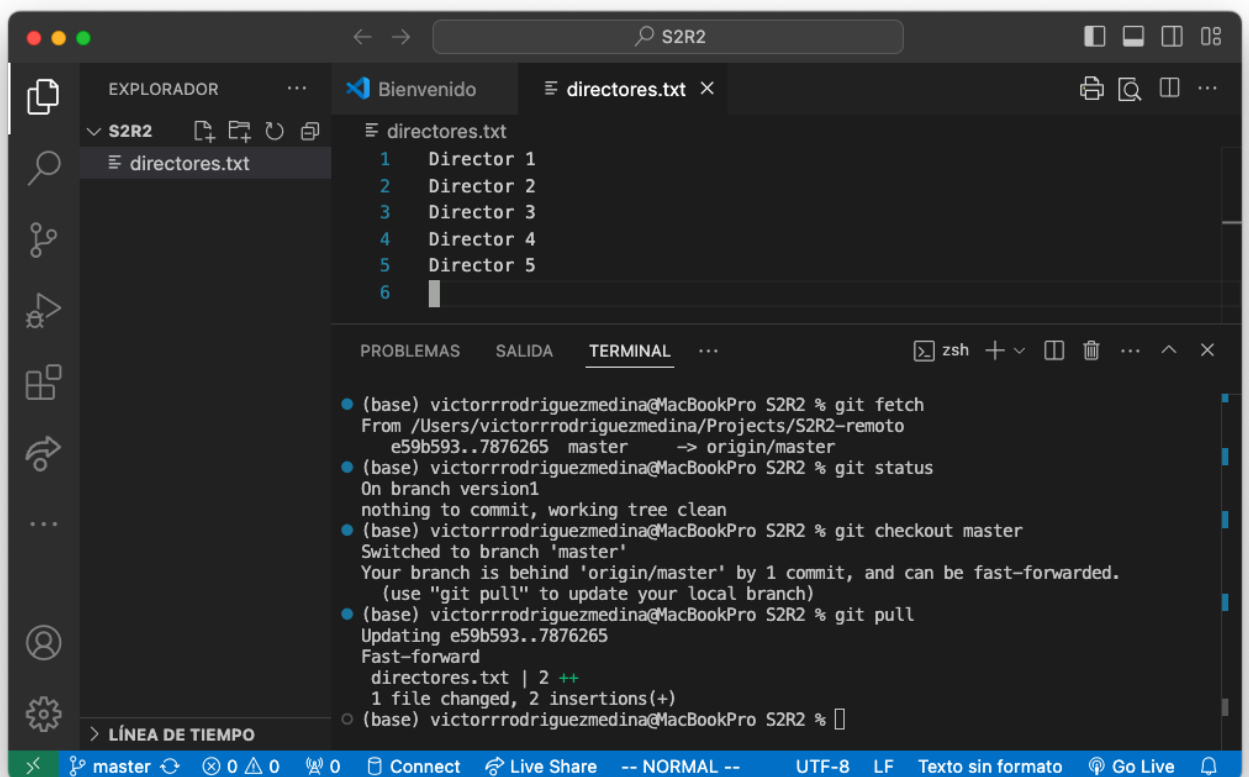
- Obtén los cambios que hay en el repositorio remoto sin fusionarlos en la rama local. Adjunta captura de pantalla del resultado del comando utilizado.



The screenshot shows the Visual Studio Code editor with a file named `directores.txt` open. The file contains a list of five directors: Director 1, Director 2, Director 3, Director 4, and Director 5. The terminal window at the bottom shows the following commands and output:

```
(base) victorrrodriguezmedina@MacBookPro S2R2 % git fetch
From /Users/victorrrodriguezmedina/Projects/S2R2-remoto
e59b593..7876265  master    -> origin/master
o (base) victorrrodriguezmedina@MacBookPro S2R2 %
```

- b) Actualiza la rama master local con el contenido de la rama master del repositorio remoto. Adjunta captura de pantalla del resultado del comando utilizado.
- c) Comprueba que aparecen los 5 nombres de directores esperados.



The screenshot shows the Visual Studio Code editor with the `directores.txt` file open. The terminal window shows the following commands and output:

```
(base) victorrrodriguezmedina@MacBookPro S2R2 % git fetch
From /Users/victorrrodriguezmedina/Projects/S2R2-remoto
e59b593..7876265  master    -> origin/master
(base) victorrrodriguezmedina@MacBookPro S2R2 % git status
On branch version1
nothing to commit, working tree clean
(base) victorrrodriguezmedina@MacBookPro S2R2 % git checkout master
Switched to branch 'master'
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)
(base) victorrrodriguezmedina@MacBookPro S2R2 % git pull
Updating e59b593..7876265
Fast-forward
 directores.txt | 2 ++
 1 file changed, 2 insertions(+)
o (base) victorrrodriguezmedina@MacBookPro S2R2 %
```



## 4. GitHub

1. Crea una cuenta en GitHub

2. Añade tu dirección de correo de educación

3. Solicita un descuento para uso educativo: <https://education.github.com/pack>

4. Haz un fork del repositorio localizado en la siguiente url: <https://github.com/curso-github-cefire/sesion3-practica>. A partir de este momento todas las tareas que se indican se deben realizar en tu repositorio (el que has clonado mediante el fork).

a) Realiza un primer commit para poner tu nombre y apellidos en el fichero README.md

b) Crea 3 issues con los siguientes títulos. Si no ves la pestaña de issues, actívala desde los ajustes (settings) del repositorio.

- Añadir 3 libros
- Añadir 3 películas
- Añadir 3 discos

c) Crea una milestone denominada Tareas sesión 3-2 que contenga los 3 issues creados.

d) Modifica los ficheros correspondientes y realiza 3 commits para realizar cada una de las tareas que se indican en los issues. El mensaje del commit debe hacer que se cierren los issues correspondientes de manera automática.

e) Haz una captura de pantalla de los comandos que has utilizado para hacer los commits y subir los cambios a GitHub.

```
● (base) victorrrrodriguezmedina@MacBookPro sesion3-practica % git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   libros.txt

● (base) victorrrrodriguezmedina@MacBookPro sesion3-practica % git commit -m "Libros. Close #1"
[master 1b48d66] Libros. Close #1
 1 file changed, 4 insertions(+)

● (base) victorrrrodriguezmedina@MacBookPro sesion3-practica % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 360 bytes | 360.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/vicrodmed/sesion3-practica.git
```

```
● (base) victorrrrodriguezmedina@MacBookPro sesion3-practica % git commit -m "Películas Close #2"
[master 944ef7f] Películas Close #2
 1 file changed, 4 insertions(+)

● (base) victorrrrodriguezmedina@MacBookPro sesion3-practica % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 370 bytes | 370.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/vicrodmed/sesion3-practica.git
 1b48d66..944ef7f master -> master
```

```
● (base) victorrrodriguezmedina@MacBookPro sesion3-practica % git commit -m "Discos Close #3"
[master 447aa64] Discos Close #3
 1 file changed, 3 insertions(+)
● (base) victorrrodriguezmedina@MacBookPro sesion3-practica % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 346 bytes | 346.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/vicrodmed/sesion3-practica.git
 944ef7f..447aa64  master -> master
```

f) Incluye las capturas de pantalla en el repositorio dentro de la carpeta capturas. Añádelas también al repositorio de manera que queden guardadas en tu repositorio en GitHub.

g) Realiza una pull request indicando en el mensaje que has completado la tarea.